Blockchain : Introduction and Recent Developments in Game Theory

Louis Bertucci

Institut Louis Bachelier
University of California, Berkeley

March 13, 2020

Roadmap

1. Blockchain : Origins
2. The Consensus Problem
3. The Bitcoin Blockchain : Proof-of-Work
4. MFG Approach to Bitcoin Mining

1 What is a blockchain?

At the intersection of many sciences

- Cryptography
- Distributed Systems
- Economics / Finance

Focus of this talk

- A lot of aspects/use cases/tech in blockchain
- But today, I'll focus on consensus (the core of a blockchain)

Cypherpunk Movement

- Active movement since the late 1980s
- Cryptography and privacy-enhancing tech
- They value privacy... a lot
- Cypherpunk manifesto (1993) :
  - "Privacy is necessary for an open society in the electronic age"
  - "Privacy in an open society requires anonymous transactions systems"
- development of many modern encryption / communication protocols (PGP, BitTorrent, Wikileaks, Tor...)

Attempts to build P2P "E-Cash"

- DigiCash (1989)
- BitGold (1998)
- PayPal, Venmo, Revolut, etc. (not P2P)

But **double spending** is the hard part : cash-like behavior
$\Longrightarrow$ all rely at some point on a central authority

Solutions

- need a decentralized accounting system
- hard because of the consensus problem

2 Distributed Consensus

Consensus

- require a set of agents (processes) to agree on the value of a system (binary value or full database)
- agents cannot "talk" to each other, they must use a message system

Impossibility Results

1. Synchronous consensus (Byzantine General Problem)
2. Asynchronous consensus

Synchronous - The Byzantine Generals Problem

- an enemy city is surrounded by n generals of the byzantine army
- $m < n$ generals are traitors
- All honest generals must agree on : Attack / Retreat
- Message system : Synchronous - the absence of a message can be detected

Lamport, Shostak and Pease (1982) - Byzantine Fault Tolerance

- In a synchronous message-passing distributed system with m malicious generals, consensus is only possible with $n \geq 3m + 1$ generals

Asynchronous Consensus

- n processes must come to an agreement
- all processes are honest (no Byzantine-like behavior)
- processes might be faulty : crash after a finite number of steps
- Message system : Asynchronous - cannot differentiate between a dead process and a very slow process

Fischer, Lynch and Paterson (1985) - Asynchronous Consensus

- In a fully asynchronous message-passing distributed system, consensus is impossible with one faulty process

The Bitcoin Whitepaper

On a Cryptography mailing list, an email is sent by "Satoshi Nakamoto"

Bitcoin P2P e-cash paper

2008-10-31 18:10:00 UTC - Original Email - View in Thread

```
I've been working on a new electronic cash system that's fully
peer-to-peer, with no trusted third party.

The paper is available at:
http://www.bitcoin.org/bitcoin.pdf
```

```
The main properties:
Double-spending is prevented with a peer-to-peer network.
No mint or other trusted parties.
Participants can be anonymous.
New coins are made from Hashcash style proof-of-work.
The proof-of-work for new coin generation also powers the
network to prevent double-spending.
```

$\Longrightarrow$ How is it possible given impossibility of consensus ?

The Blockchain Consensus

1. "Consensus is impossible"
2. Add economic incentives to that
3. Consensus becomes (asymptotically) possible

3 The Bitcoin Proof-of-Work Protocol

The Bitcoin Protocol

protocol allowing a group of untrusted agents to reach consensus over the content of a distributed database, called the blockchain

- the blockchain contains the list of every Tx, grouped into blocks and chained together
- validation of new blocks does not rely on any trusted party
- validators (called miners) take turn creating new blocks

Prerequisite - Hash Functions

- Hash function (H) : $\{0,1\}^* \to \{0,1\}^k$
- Cryptographic Hash functions :
  - Fast to compute
  - No-prediction : cannot predict $H(m)$ without running the algorithm on m
  - Pre-image resistant : difficult to find m from $H(m)$
  - Collision resistant : difficult to find $m_1 \neq m_2$ st $H(m_1) = H(m_2)$

Exemple : SHA-256

- $k = 256$
- $sha256$('hello, world') $= 0x09ca7e4e...08360d5b$
- $sha256$('hello, world!') $= 0x68e656b2...f368f728$
- looks random

Hash-based Puzzle

- Assume $H : \{0,1\}^* \longmapsto \{0,1\}^{256}$
- $P(H(x) \in \{\{0\} \wedge \{0,1\}^{255}\}) = 0.5$
- $P(H(x) \in \{\{0\}^2 \wedge \{0,1\}^{254}\}) = 0.25$
- ...

Proof-of-Work

- given a target it is possible to prove the work done
- no other approach than brute-force

Block Creation

- block valid if enough proof-of-work (ie. must come with a hash below a target)
- data hashed :
  - block info : timestamp, hash of transactions
  - hash of previous block
  - *NONCE*

An ever growing list of valid blocks

Dynamic Difficulty Adjustment

- for stability reasons, the average inter-block time must be constant (10 min on Bitcoin)
- the target for the hash dynamically adjusts
- part of the consensus rules

Consensus Rules - Block valid if :

- all transactions are valid
- no double spending
- proof-of-work
- ...

Economic Incentives

Reward

- deterministic coin issuance given out as reward for miners
- bitcoin issuance sheme : 50 BTC for genesis block then halved every 210,000 blocks ($\sim$ 4 years)
- optional transactions fees

Cost

- as more miners enter the network, the hash target decreases
- need a lot of hardware to compute hashes and a lot of electricity (power and cooling systems)

Race

- price of bitcoin gives miners incentives to mine **valid** blocks
- economic incentives create a race among miners

Security of the Blockchain

Longest Chain Rule

- anyone can trust the longest chain as being approved by the majority of miners (hashrate)
- largest Proof-of-Work

Security

- if an attacker wants to change a Tx which happened 1k blocks ago
- he needs to recompute all the child blocks PoW faster than the honest nodes grow the honest chain (51% attack)
- as the hashrate grows, it becomes more and more expensive to sustain such an attack

$\Longrightarrow$ as Tx get buried deep into the blockchain, they become part of consensus

A solution to consensus (asymptotically)

- The blockchain can be "trusted" because of the amount of proof-of-work
- Consensus being solved (asymptotically) means double spending is solved without trusted third party
- this also means that a decentralized accounting system is available
- and that a P2P e-cash system works

MFG Approach to Bitcoin Mining

Louis Bertucci Charles Bertucci Jean-Michel Lasry
Pierre-Louis Lions

Institut Louis Bachelier
CNRS
Université Paris-Dauphine
Collège de France

Introduction

- The total hashrate represents the security of the blockchain
- Equilibrium behavior of the total hashrate
- Proof-of-Work fits very well in the MFG framework :
  - a lot of non-atomic players
  - interacting only through the aggregate computational power

Historical Hashrate

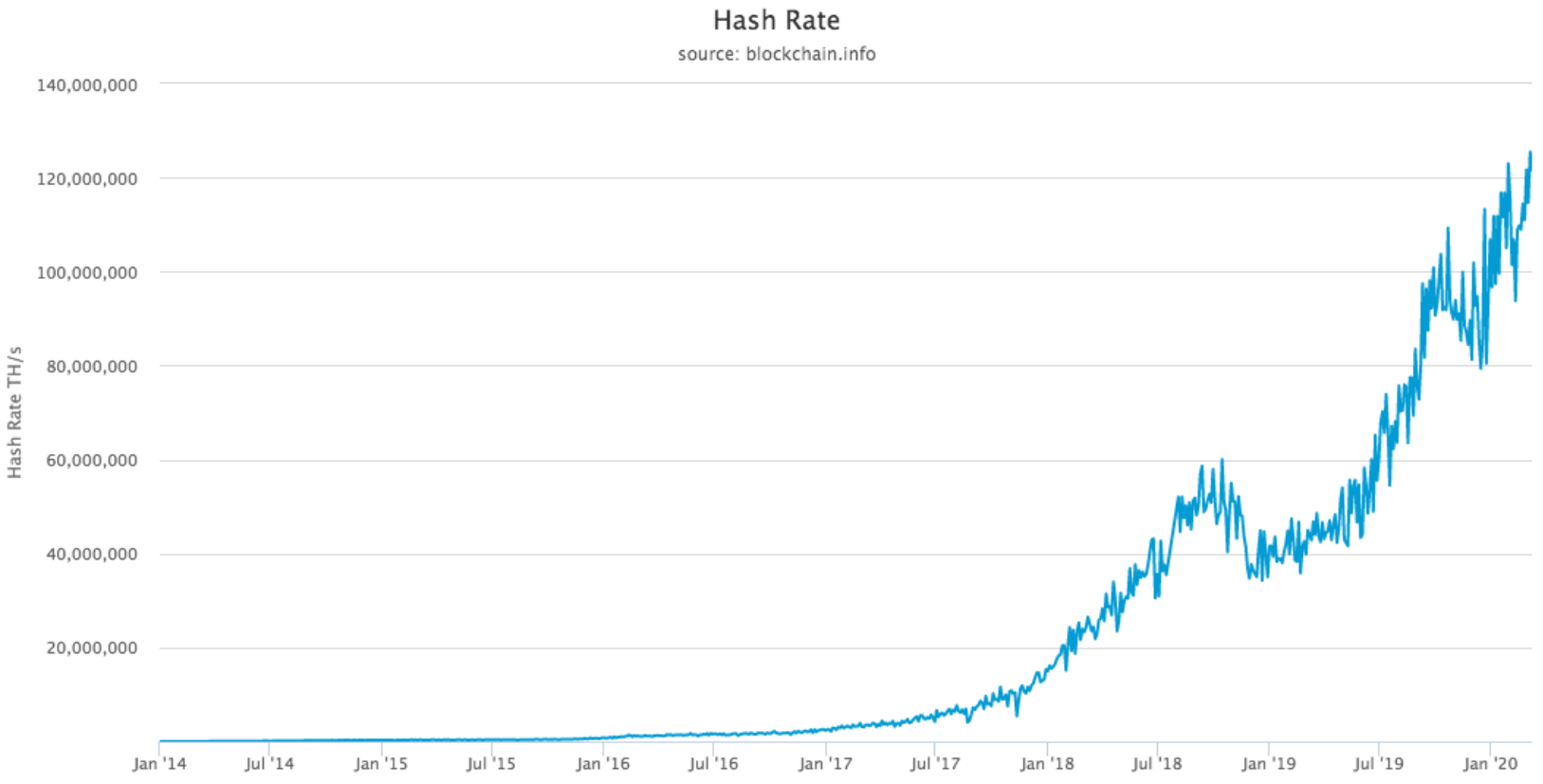

Figure : Total Bitcoin hashrate since January 1st 2014

Model Setup

Model assumptions

- Constant cryptocurrency price
- Perfect mining diversification
- Sticky investment

Technological progress

- miners demand for high-performance mining computer chips generates technological progress
- assume constant technological progress rate : δ
- "real hashrate", K_t, is the "nominal hashrate", P_t, discounted by δ

$$K_t := e^{-\delta t} P_t$$

$\Longrightarrow$ focus on the value of 1 unit of real hashrate, denoted by U

Miners Optimization Problem

- miners discount time at rate r
- sticky investment $\beta \in \mathbb{R}_+$
- Collective miners problem

$$\max_{X_t} \int_0^{+\infty} e^{-rt} \left(X_t U_t - \beta \left(\frac{dX_t}{dt} \right)^2 \right) dt$$

- Optimal decision

$$dX_t = \frac{1}{2\beta} U_t dt = \lambda U_t dt$$

- The dynamics of the real hashrate is therefore

$$\dot{K}_t = -\delta K_t + \lambda U_t$$

$\Longrightarrow$ need to understand the value of the real hashrate U_t

Value of a Unit of Real Hashrate

- denote by c the electricity cost associated to running the machines
- Value function :

$$U(K) := \int_0^\infty e^{-(r+\delta)t} \left(\frac{1}{K_t + \epsilon} - c \right) dt$$

where $(K_t)_{0 \leq t}$ is the process satisfying

$$\begin{cases} dK_t = -\delta K_t dt + \lambda U(K_t) dt \\ K_0 = K \end{cases}$$

- If U is smooth it satisfies the master equation

$$0 = -(r+\delta)U + U'_K(-\delta K + \lambda U) + \frac{1}{K+\epsilon} - c \ \text{ in } \ [0, \infty)$$

Results (1)

Theorem 3.1

There exists a unique lipschitz function solution of the master equation. This function is decreasing and satisfies $U(0) \geq 0$ and $\lim_{K \to \infty} U(K) \leq 0$.

- The idea of the proof relies on the monotone structure of the MFG
- the block reward $\frac{1}{K+\epsilon}$ is decreasing in K
- the function that yields the derivative of the hashrate is increasing in U, i.e. $U \to -\delta K + \lambda U$

Results (2)

Proposition 3.1

There always exists a stationary state K_0 that can be computed explicitly in terms of the model parameter. Moreover,

$$\lim_{t \to \infty} K_t = K_0$$

- The stationary state is defined as

$$\dot{K}(K_0) = 0 \iff \delta K_0 = \lambda U(K_0)$$

- From the master equation we have

$$U(K_0) = \left(\frac{1}{K_0 + \epsilon} - c \right) (r + \delta)^{-1}$$

- which yields $K_0 > 0$

$$\delta K_0 = \frac{\lambda}{r + \delta} \left(\frac{1}{K_0 + \epsilon} - c \right)$$

On the use of the master equation

- the previous analyzis could have been made with the potential case : the value function of a monopolist
- the master equation allows for a more fine grained approach
- explain the behavior of aggregate quantity, $(K_t)_{t\geq 0}$, from individual decisions, λU
- modeling made "easier"
- sometimes it is impossible to use HJB

Extension : multiple machines

- Assume there are 2 types of machines : type-1 contributes $\gamma < 1$ to the total
- Both can be used for an external activity (AI, rendering, renting)
- We obtain a system of master equations

$$\begin{cases} 0 = -(r+\delta_1)U + \partial_{K_1} U(-\delta_1 K_1 + \lambda_1 U) + \partial_{K_2} U(-\delta_2 K_2 + \lambda_2 V) \\ \qquad\qquad + \max\left\{ \frac{\gamma}{\gamma K_1 + K_2 + \epsilon} - c_1; \alpha_1 \right\} \quad \text{in} \quad [0,\infty)^2; \\ 0 = -(r+\delta_2)V + \partial_{K_1} V(-\delta_1 K_1 + \lambda_1 U) + \partial_{K_2} V(-\delta_2 K_2 + \lambda_2 V) \\ \qquad\qquad + \max\left\{ \frac{1}{\gamma K_1 + K_2 + \epsilon} - c_2; \alpha_2 \right\} \quad \text{in} \quad [0,\infty)^2; \end{cases}$$

Results (2)

Theorem 5.1

There exists a unique lipschitz couple (U, V) solution of the system of master equations such that $U(0, \cdot) \geq 0$ and $V(\cdot, 0) \geq 0$. Moreover this couple satisfy the monotonicity property :

$$\begin{pmatrix} D_x U & D_y U \\ D_x V & D_y V \end{pmatrix} \text{ is positive-definite}$$

together with the fact that both U and V are decreasing with respect to x and y.

Proposition 5.1

There exists a unique stationary state (x_0, y_0) and all induced trajectories converge toward it.

$\Longrightarrow$ Here as well, the proofs heavily relies on the monotonic structure of the MFG.

Thank You !

Blockchain : Introduction and Recent Developments in Game Theory

Louis Bertucci

Institut Louis Bachelier
University of California, Berkeley

March 13, 2020